HW2:
PERCEPTRON.

Submit your assignments on Gradescope.
Use the provided Python template file, and complete the functions ONLY (DO NOT edit function definitions, code outside the function, or use other libraries).
Submit your solution names as **HW2.py**

This is a coding assignment.

In this assignment, we will implement the Perceptron learning algorithm.
More specifically, we are interested in seeing the weight update rule and visualizing the data.

The dataset we will use in this assignment is named *iris* (can be downloaded from the course Google Drive)

(1) Fill in the function ***extract_features_label*** that returns features and the label from the data. See the function definition (and sample main function) for input/output types.
The features we are interested in are ***sepal.length*** and ***sepal.width*** and the labels are **Setosa**: 0, **Virginica**: 1.
.

(2) Fill in the function ***fit***. You need to implement the weight and bias update rule for perceptron learning.

(3) Fill in the function ***predict.*** You need to predict the labels for the given inputs.

(4) Fill in the function ***_unit_step_func***. Implement the step threshold function.

(5) The code for visualizing the data is provided.

This is a report assignment.

Using the code you generated, and slightly modifying it, apply the following.

1) Keep the learning factor constant = 0.1 and then change the number of epochs to 10, 50, 100 and run the simulations three times.
   plot the data and three decision boundaries on the same figure.

2) Keep epoch number constant = 20, and then change the learning factor 0.01, 0.1, 0.9 and run the simulations three times.
   plot the data and three decision boundaries on the exact figure.

Comment on the results, explaining the effects of the changes in the parameters on the learning process.

Plot errors versus number of epochs. (Accumulate the error at each epoch) and describe the convergence (only one of the above cases).

3) Use sepal width and sepal length as features and Versicolor and Virginica as classes.
   Run the perceptron algorithm for learning_rate=0.01, epochs=1000.
   Plot the results. Explain why the algorithm did not achieve high accuracy.